



Citation for published version:

Day, M & Ball, A 2009, 'Institutional repositories should be built on open source software - negative argument', Bulletin of the American Society for Information Society and Technology, vol. 35, no. 4, pp. 22-26.

Publication date:
2009

[Link to publication](#)

University of Bath

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Topic 3: Institutional repositories should be built on open source software - negative argument

Michael Day and Alexander Ball¹
UKOLN, University of Bath
Bath BA2 7AY, United Kingdom
{m.day, a.ball}@ukoln.ac.uk

A quick glance at the most recent statistics produced by the *OpenDOAR* Directory of Open Access Repositories suggests that the vast majority of existing institutional repositories are currently built upon open source software. For example, the tables show that at the end of January 2009, almost half (47 percent) of the repositories listed in the directory use one of the two leading open source repository packages [1]. While this demonstrates that there is certainly a market for open source repository software, it does not necessarily follow that *all* repositories should be built upon it. To argue this is *not* to suggest that there is anything fundamentally wrong with the open source development model itself. The open source philosophy has proved itself to be a very successful model for software development. It has also been a major inspiration for the collaborative models that underpin many recent Internet developments as well as for the concept of open science [2]. In the institutional repository context, however, there are a number of reasons why an insistence on open source software solutions may not be strictly necessary.

The first reason relates to the ever-changing technical context of repositories. Clifford Lynch's definition of institutional repositories emphasizes that they are not "simply a fixed set of software and hardware" [3]. While at any given time repositories will have to be supported by a set of technologies, Lynch argues that they essentially constitute an organizational commitment to the ongoing stewardship of the digital content created by an institution and its members, and that a key service will be "the management of technological changes, and the migration of digital content from one set of technologies to the next." Even where institutions have motives other than long-term stewardship for setting-up repositories, it remains the case that the technical aspects of systems will need to evolve through time to take account of changes in institutional policies and requirements and to take advantage of the functionality offered by the latest software platforms and tools. In this rapidly

¹ This is a draft version of an essay produced for a special section on "Institutional Repositories: The Great Debate," guest edited by Helen R. Tibbo, Rachael Green Clemens, and Carolyn Hank (School of Information and Library Science, University of North Carolina at Chapel Hill) for the *Bulletin of the American Society for Information Society and Technology*. It was published together with an opposing viewpoint provided by Paul Jones of UNC Chapel Hill in: *Bulletin of the American Society for Information Society and Technology*, vol. 35, no. 4 (April-May 2009), pp. 22-26. Retrieved May 28, 2009, from <http://www.asis.org/Bulletin/Apr-09/>

changing context, it does not make sense to limit the choice of tools to just those that happen to follow a particular software-licensing model.

Dealing with the practical aspects of repository development highlights a second set of reasons why open source software should not necessarily be seen as essential for institutional repositories. As suggested above, the technical choices that need to be made by repository managers should be grounded firmly in institutional requirements. The questions that institutions need to be ask include, for example: whether it would be possible to integrate (or develop) other tools within the chosen software framework; whether the system - when developed - would be able to interoperate with all relevant systems, both internal and external; and whether it would be possible to get content (and its associated metadata) in and out of the system easily. The answers to these questions - primarily focused on the consistent use of standards and application programming interfaces (APIs) - should be far more important than the exact software development model in use. In any case, utilizing open source software does not *guarantee* that institutions will avoid the potential problems of 'vendor lock-in' or ensure that repository platforms will be either stable or sustainable. Institutions can attempt to hedge some of these technical questions either by co-operating with other institutions or by contracting out repository development and/or hosting to specialist organizations. A growing number of subscription-based services are now emerging that aim to provide institutions with repository services, with options based on both open source and proprietary software. Whatever repository development choices are made, however, it will be necessary to ensure that systems do not become dead-ends. This will be dependent on the appropriate use of standards. For example, in their paper on the outsourced University of Wollongong repository, Organ and Mandl have pointed out that one of their key principles "was to deploy a repository consistent with a range of standards so that material loaded could be transferred as necessary at a later date to a different system" [4].

A third set of reasons why open source software should perhaps not be viewed as the only acceptable approach to institutional repository development relates to the nature of the open source process itself. Open source software, by its very nature, tends to be developer driven. In itself, this need not be a problem. However, in the repository domain, this *can* result in a mismatch between specific institutional requirements and what software is actually able to provide at a given moment of time. While in an ideal open source context, collaborative community development would be able to fill gaps and resolve many of the other potential conflicts, the anecdotal examples provided by Dorothea Salo in her recent article on institutional repositories suggests that the current situation is far from perfect. While recognizing many of their benefits, she comments that the three main open source software offerings currently "offer varying quantities of installation and maintenance headaches, expensive hardware demands, customization and development hassles, and poor fit with existing library software, websites, and services" [5]. Similarly, a 2007 report for UNESCO's Memory of the World programme suggested that one of the major open source repository platforms "has evolved into a monolithic software application, and complex code base, that introduces potential scaling and capacity constraints for some large institutional users" [6]. While it is fair to point out that these problems are certainly not unique to open source, it may be a signal that software development in the repository domain is currently immature. Certainly the rapid development cycles

typical of open source software can make the local customization of repositories problematic. Time spent carefully redesigning repository interfaces to meet local needs can be wasted when updated versions of repository platforms are released. Solutions might include the modularization of repository platforms combined with the promulgation of consistent and stable standards and APIs.

In her article, Salo suggests that some repository software platforms need to be able to be more responsive to specific institutional requirements, for example with regard to things like mediated deposit or the batch import of documents. There may also be a need for repositories to interact more closely with a wealth of other institutional systems, which are currently typically based on a mix of proprietary and open source solutions. While a recent report suggests that institutions in the US (at least) might welcome additional open source development within the higher education sector [7], it might seem perverse in institutional terms to insist that repositories require an open source solution, while course management or library management systems are free to follow the proprietary path.

Finally, and perhaps most importantly, the insistence that institutional repositories should always be built on open source software - regardless of context - would seem to be unnecessarily focused on the means rather than the ends. The purpose of any repository should be the stewardship of well-managed collections of institutional content. Therefore, any focus on 'openness' should be concentrated on making sure that repository content and its associated metadata can be exposed to other systems through tools like the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), and that both content and metadata can be exchanged successfully between repositories and other systems. In the same way that repository technologies will change over time, these interoperability mechanisms will also need to evolve to take account of new opportunities for sharing data. The experience of some data science domains suggests that there is a need to focus a great deal of attention on adherence to open standards and on the development of stable APIs, as well as on shared approaches to semantics [8].

To conclude, where open access is the main objective of an institutional repository, the exact license status of the software that underlies it does not seem particularly significant. While the statistics from *OpenDOAR* suggest that most repositories are currently developed on open source platforms, a growing market for outsourced solutions exists, including for those provided by the commercial sector. In the longer term, however, things could become even more complex. For example, institutions could contract-out some 'core' repository functions to third party services based in the 'cloud' [9]. Simultaneously, however, repositories are also likely to depend increasingly on their tighter integration within a more complex set of institutional systems and processes (e.g. as part of research workflows), and in many cases linked to national and international research e-infrastructures. The open source development model is likely to have a very significant role to play in helping to develop and link these complex infrastructures, but other approaches will still remain viable.

References

1. *OpenDOAR Usage of Open Access Repository Software - Worldwide*. Retrieved January 30, 2009, from <http://www.opendoar.org/>
2. Tapscott, D., & Williams, A. D. (2008). *Wikinomics: How mass collaboration changes everything*, rev. ed. London: Atlantic Books.
3. Lynch, C. A. (2003). Institutional repositories: Essential infrastructure for scholarship in the digital age. *Portal: Libraries and the Academy*, 3(2), 327-336.
4. Organ, M., & Mandl, H. (2007). Outsourcing open access: Digital Commons at the University of Wollongong, Australia. *OCLC Systems & Services*, 23(4), 353-362.
5. Salo, D. (2008). Innkeeper at the roach motel. *Library Trends*, 52(2), 98-123. Retrieved January 30, 2009, from <http://minds.wisconsin.edu/handle/1793/22088>
6. Bradley, K., Lei, J., & Blackall, C. (2007). *Towards an open source repository and preservation system*. Paris: UNESCO. Retrieved January 30, 2009, from <http://www.unesco.org/webworld/en/mow-open-source/>
7. Courant, P. N., & Griffiths, R. J. (2006). *Software and collaboration in higher education: A study of open source software*. New York: Ithaca. Retrieved January 30, 2009, from http://www.ithaka.org/strategic-services/oss/OOSS_Report_FINAL.pdf
8. Goble, C., & Stevens, R. (2008). State of the nation in data integration for bioinformatics. *Journal of Biomedical Informatics*, 41, 687-693.
9. Lynch, C. A. (2008). A matter of mission: Information technology and the future of higher education. In R. N. Katz (Ed.), *The tower and the cloud: Higher education in the age of cloud computing* (pp. 43-50). Boulder, CL: EDUCAUSE. Retrieved January 30, 2009, from <http://www.educause.edu/thetowerandthecloud/133998>